Programación Matemática y Software (2014) 6 (1): 23-31. ISSN: 2007-3283

Firmador y autenticador de documentos digitales

Certifier of digital documents

Luis E. Cuevas-Díaz [1*], Juan C. López-Pimentel [1], Víctor F. Ramos Fon-Bon, [1] Rodolfo Ibarra-Orozco [1]

[1] Universidad Politécnica de Chiapas.
Calle Eduardo J. Selvas s/n y av. Manuel de J. Cancino, col. Magisterial. México. CP 29082
* Coreo-e: lecd89@gmail.com

PALABRAS CLAVE:

protocolos de seguridad, esteganografía, criptografía, huella digital y marcas de agua

RESUMEN

Garantizar que un documento sea auténtico no es una tarea fácil y menos si éste es digital. Actualmente existen muchas herramientas computacionales que permiten falsificar documentos. El objetivo de la investigación descrita en este artículo consistió en desarrollar un servicio web que permitiera incluir mecanismos de seguridad a documentos digitales y con eso garantizar su autenticidad. Para proteger este sitio, se dividieron los procesos en nivel *host* y nivel *network*. El primero tiene que ver con los procesos internos tanto del cliente como del servidor, lo cual involucra aspectos de seguridad computacional como esteganografía y criptografía. El segundo nivel tiene que ver con la protección del flujo de información que viaja del cliente al servidor, para ello se desarrolló un protocolo de seguridad basado en criptografía simétrica, que se ha verificado formalmente para tener la seguridad de que matemáticamente se parte de un protocolo seguro. Este artículo reporta los fundamentos técnicos para implementar una aplicación web que permita firmar y autenticar documentos digitales, y a esta aplicación se le ha denominado FADD.

KEYWORDS:

security protocols, steganography, cryptography, digital signature and watermarking

ABSTRACT

Ensure the authenticity of a document is not an easy task. Currently there are many computational tools allowing forging documents. The objective of this research was to develop a web service allowing including security mechanisms to digital documents and thereby ensuring its authenticity. To protect the web service, it was split in two levels: host and network. The first level has to do with the internal processes of both the client and server; this involves computer security aspects such as steganography and cryptography. The second level is related with protecting the flow of information traveling from client to server, to do this it has been developed a security protocol based on symmetric cryptography, which has been verified formally. This paper reports the technical foundation to implement a web server that allow authentication in a digital document, which have been called FADD.

1 INTRODUCCIÓN

La digitalización provee grandes ventajas, entre ellas: a) evita el gasto inmoderado de papel y por ende evita la destrucción de bosques y daños inminentes al medio ambiente,† y b) permite una mejor administración de documentos. Sin embargo, también tiene algunas desventajas, en particular la falsificación de documentos. Esta investigación se concentró en proveer mecanismos de seguridad en documentos digitales que permitan:

- *Proveer protección de derechos de autor en documentos digitales.*†† Muchos investigadores agregan información redundante en el contenido de un documento digital aplicando esteganografía (proceso que consiste en agregar información oculta a un documento), tales como autor, fecha, dirección, etcétera. Cuando este proceso se usa para autenticar un documento se le conoce como marca de agua, en inglés *watermarking*††† [3]. Éste se usa para esconder información del propietario sobre un medio digital como fotografías, música, videos, documentos [4], el objetivo es proteger el contenido de interés.
- *Verificar la autenticidad de un documento digital previamente firmado*. Este proceso consiste en recuperar información que incluye la firma del documento, o lo que es usado para proveer evidencia de originalidad.

Para desarrollar el sitio web se debe proteger el flujo de información y los procesos locales, es por ello que se dividieron los procesos en dos niveles: *host* y *network*. El primer nivel se relaciona con los procesos internos tanto del cliente como del servidor, esto involucra aspectos de seguridad computacional tales como esteganografía y criptografía. El segundo nivel involucra proteger el flujo de información que viaja del cliente al servidor, para ello se desarrolló un protocolo de seguridad basado en criptografía simétrica, el cual se verificó formalmente para tener la seguridad de que matemáticamente se estuviera partiendo de un protocolo seguro [5].

Hay algunos trabajos muy relacionados con éste [6-11], pero se notó que ninguno mencionó aspectos de implementación y tampoco verificaron formalmente el protocolo base.

Este artículo se organiza de la siguiente manera: la sección 2 introduce los detalles técnicos de los dos niveles arriba mencionados, como son esteganografía, *watermarking*, criptografía y protocolos de seguridad. En la sección 3 se describe el protocolo de seguridad base para la implementación. La sección 4 describe el firmador y autenticador de documentos digitales (FADD) en el nivel escritorio y en el nivel web, así como algunos problemas encontrados en la fase de desarrollo de este proyecto. La sección 5 describe algunos trabajos que están íntimamente relacionados con el trabajo que se desarrolló. Finalmente, la sección 6 describe las conclusiones y trabajo futuro.

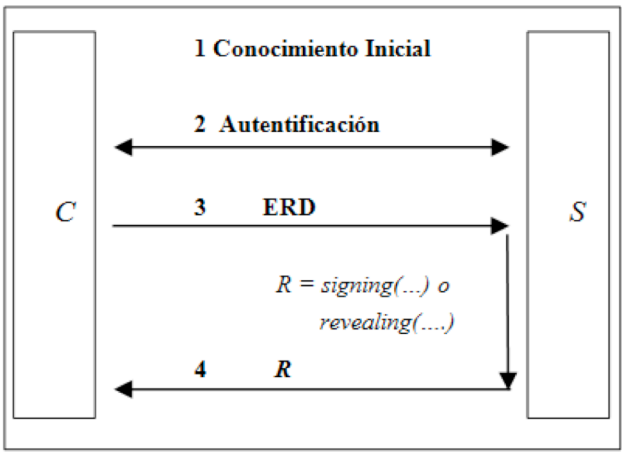

Figura 1. Validación cruzada K=3

2 DETALLES TÉCNICOS EN LOS NIVELES *HOST* Y *NETWORK*

La figura 1 ilustra el proceso de firma y certificación en un contexto en red, *S* representa el sitio web (servidor) y *C* al cliente. La flecha denota un flujo de datos y una flecha doble significa intercambio de datos entre los participantes. La flecha hacia abajo denota un proceso local. Como se puede ver, existen dos partes: a) *host* y b) *network*.

Host involucra el conocimiento inicial que *C* y *S* poseen antes de interactuar, el proceso de firma y

† Entre los años 1960 y 2000 la demanda de productos provenientes de ecosistemas creció de forma significativa al duplicarse la población mundial, la tala de árboles para pulpa y producción de papel se triplicó [1]. De acuerdo con los indicadores de Biodiversidad y bosques del WorldwatchInstitute 40,000 hectáreas de bosques desaparecen cada día a causa de la creciente demanda de madera, celulosa y papel [2].

†† Hemos utilizado el término documento digital o e-document indistintamente.

††† Se utiliza el término marcas de agua o *watermarking* indistintamente.

verificación de documentos digitales que realiza *S* como proceso local, en la figura 1 esto es *signing*( ) o *revealing*( ); y las actividades que *C* pueda hacer como proceso local.

Network involucra tres principales procesos: i) autenticación; ii) envío y recepción de datos (ERD), y iii) envío del resultado de acuerdo con el tipo de proceso que se está solicitando, *R* denota envío de documento digital autenticado si es aplicada la función *signing*( ), en caso de aplicarse *revealing*( ) entonces *R* contiene los datos de la firma.

2.1 Conceptos y formalismo en el nivel *host*

Para realizar la firma en un documento digital y su respectiva consulta se aplican técnicas de criptografía y esteganografía.

Criptografía. Se define como el estudio de sistemas matemáticos que involucran dos tipos de problemas de seguridad: privacidad y autenticación [12].

- *Privacidad*. Previene que la información transmitida sobre un canal público sea extraída por usuarios no autorizados.
- *Autenticación*. Se relaciona usualmente con dos términos: autenticación de mensaje y de usuario. El primero consiste en verificar que un mensaje fue enviado por quien reclama ser el editor. El segundo consiste en verificar que un agente es quien reclama ser.

En general, la criptografía tiene dos principales mecanismos: llave simétrica y asimétrica. Por el momento, se discutirá sólo el mecanismo de llave simétrica.

Una llave simétrica será representada con la notación K_{AB}, donde *A* y *B* denotan los agentes que comparten la llave. El símbolo {| *m* |}*K* denota que un mensaje *m* se ha cifrado bajo la llave *K*.

Esteganografía. Se define como el arte de encapsular datos o esconder datos [13]. Muchos autores la categorizan como una forma de criptografía debido a sus similitudes, pero en realidad son diferentes [14]. El proceso de esteganografía involucra dos partes: el mensaje que se desea esconder y el transportador (*carrier* en inglés) [14]. El *carrier* es un archivo (que tiene un formato específico tales como jpg, bmp, png, pdf, mp3, etcétera), donde uno quiere esconder un mensaje. La esteganografía provee aplicaciones importantes en el mundo digital, la más notable es *watermarking*; usado principalmente para identificación de documentos y esto involucra introducir información en un *carrier* sin cambiar la apariencia (al menos no visible para el ojo humano). De acuerdo con Potdar y colaboradores [15], una de las principales aplicaciones de *watermarking* es protección de derechos de autor, administración de contenido, inserción de metadatos, detección de anomalías, huella digital y monitoreo de *broadcast*. Este trabajo se enfocó en la inserción de metadatos y detección de anomalías.

Dado que *F* es un *e-document* (el *carrier*) y *m* es cualquier mensaje que se desee ocultar con fines de *watermarking*. Después de un proceso de esteganografía se obtiene *F´*, que satisface la siguiente propiedad:

$$F \neq F' \wedge F \sim F'$$

El operador ~ denota que *F* es igual a *F´* de acuerdo con la vista humana en promedio; en este sentido, ellos son similares.

Dado que *K* es una llave simétrica, *al* es un algoritmo de esteganografía, y {*dp*} la información que servirá de firma; esto es, la información que se ocultará, la siguiente fórmula obtuvo la firma del documento usando los datos previos:

$$F' = \text{signing}(F, \{dp\}, al, K)$$

Ahora bien, para conocer quién ha firmado un *e-document*, se usa la siguiente fórmula:

$$\{dp\} = \text{revealing}(F', al', K)$$

Donde *al´* es el algoritmo inverso de *al* y *F´* denota el nuevo *e-document*, el cual satisface lo siguiente:

$$F \neq F' \wedge F \sim F'$$

Así, una vez que {*dp*} fue analizado por el cliente, se pudo conocer quién ha firmado el documento, y con ello aceptar o rechazar su autenticidad.

En la sección 4 se describen los aspectos de implementación de estas fórmulas.

2.2 Conceptos y formalismo del nivel *network*

Cualquier mensaje no cifrado transmitido por la red puede considerarse inseguro. En el caso de este servicio web, fue esencial considerar que atacantes activos y pasivos estarían siempre presentes sobre la red capaz de manipular los mensajes transmitidos. Por consiguiente, fue prioritario implementar túneles seguros o protocolos de seguridad.

Un túnel seguro, también llamado red privada virtual, en inglés *virtual private network* (VPN), encapsula los datos transferidos entre dos o más agentes; sin embargo, el principal problema es obtener autenticación; eso significa que los participantes deben autenticarse antes de que se establezca la VPN. Por el momento, en este artículo no se aborda este tipo de solución.

Por otro lado, un protocolo de seguridad es un conjunto de reglas y convenciones por donde dos o más agentes obtienen acuerdos sobre la identidad uno del otro, usualmente terminando en la posesión de uno o más secretos [16]. Estas reglas son usualmente cifradas bajo alguno de los esquemas presentados en la sección anterior.

Los protocolos de seguridad consisten de sólo unos cuantos mensajes, pero asombrosamente, muy difíciles de ser diseñados correctamente (ver a López-Pimentel y Monroy [17] para una encuesta sobre verificación de protocolos de seguridad). Por ejemplo, la detección de una falla en tres mensajes del protocolo de llave pública Needham-Schroeder (NSPK) tomó casi 17 años [18]. Más ejemplos de protocolos con fallas pueden ser consultados en la biblioteca Clark-Jacob† y también en la AVISPA.††

La notación presentada en la tabla 1 fue utilizada para la representación informal de un protocolo se seguridad.

Tabla 1. Notación en el nivel *network*

Notación	Descripción
$A : ik$	Conocimiento inicial: describe que al inicio del protocolo el agente *A* conoce *ik*.
$n.\ A \rightarrow B: m$	Envío y recepción de mensajes: en el paso *n* el agente *A* envía el mensaje *m* al agente *B*, el cual *B* recibe.
$n2.\ A \rightarrow B: m;\{n\}$	Concatenación: en el paso *n2* el agente *A* envía el mensaje *m;{n}* al agente *B*, el cual *B* recibe; aquí, el símbolo ";" denota concatenación de mensaje.
$n3.\ A \rightarrow B: m$ $m2 = f(m)$ $n4.\ A \rightarrow B: m2$	Representación del proceso local en un flujo de mensaje: en el paso *n3* el agente *A* envía el mensaje *m2* al agente *B*, el cual *B* recibe; el mensaje *m2* fue generado vía un proceso local, *f(m)*, se ejecutó entre el paso *n3* y *n4*.

Tabla 2. Propuesta de protocolo de criptografía simétrica para firmar y autenticar documentos digitales

Etapa	Proceso	Descripción
I	Conocimiento inicial $C : K_{CS}, un$ $S : Key, K_{CS} \in Key, un \in AE$	El servidor debe conocer al inicio todas las llaves y algoritmos de esteganografía previamente seleccionados por el cliente.
II	Autenticación 1.- $C \rightarrow S : C; NC$ 2.- $S \rightarrow C : \{\|N_C;K'_{CS};S;NS\|\}Kcs$ 3.- $C \rightarrow S : \{\|NC;NS\|\}K'cs$	La fortaleza del protocolo consiste en que cada cliente debe mantener en secreto su llave que comparte con el servidor.
III	Firma 4.- $C \rightarrow S : \{\|F;\{dp\}\|\}_{K'cs}$ $F' = signing(F, \{dp\}, al, K_{CS})$ 5.- $S \rightarrow C : \{\|F';Hash(F)\|\}_{K'cs}$	*F = File* denota un *documento-e* y *{dp}* es un conjunto de datos personales. *F´* es un archivo firmado bajo la llave K_{CS} y usando el algoritmo *al*.
IV	Proceso de consulta segura 4.- $C \rightarrow S : \{\|F;un\|\}_{K'cs}$ $\{dp\} = revealing(F', al', K_{CS})$ 5.- $S \rightarrow C : \{\|\{dp\};Hash(F')\|\}_{K'cs}$	El servidor asocia *un* (nombre de usuario de cualquier cliente) con su algoritmo y llave respectiva.

3 PROTOCOLO DE SEGURIDAD BASE DE LA IMPLEMENTACIÓN

Para la aplicación web, se consideró que sólo clientes autenticados deben tener la posibilidad de incluir una firma en un documento. Para eso, se necesita de dos agentes, el servidor *S* y el cliente *C*. Este último solicita una conexión segura al servidor *S*. El protocolo completo se puede ver en la tabla 2, que muestra cuatro etapas: conocimiento inicial, autenticación, firma y consulta.

3.1 Conocimiento inicial y autenticación

Conocimiento inicial. Representa el conocimiento que cada agente posee al inicio de la corrida de un protocolo. En esta etapa el cliente debe previamente estar registrado con el servidor. Eso significa que sólo usuarios autorizados podrán firmar y consultar firmas.

† Clark-Jacob library está disponible en http://www.lsv.ens-cachan.fr/spore/index.html

†† AVISPA library está disponible en http://www.avispa-project.org

En el registro, cada cliente debe también especificar su algoritmo de esteganografía.

Autenticación. Describe una corrida del protocolo, donde *C* empieza una corrida con *S*. Idealmente, en el paso 1, *C* empieza una corrida del protocolo con *S* enviándole texto plano que contiene una identificación, denominada nombre de agente *C* y un *nonce* NC.

Una vez recibido el mensaje, *S* le regresa a *C* un mensaje cifrado, el cual sólo *C* puede descifrar, eso contiene el *nonce* NC (como un desafío de respuesta), una nueva llave de sesión K'_{CS}, su nombre de agente *S* como una prueba de identidad y un nuevo *nonce* NS como un desafío.

Una vez interpretado el mensaje del paso 2, *C* devuelve (en el paso 3) un mensaje cifrado bajo la nueva llave de sesión K'_{CS} y enviándole el desafío y su *nonce* original; este mensaje debe ser interpretado por *S* con una prueba de autenticación. *C* y *S* están ahora listos para comunicaciones futuras usando la llave distribuida.

Esta versión del protocolo fue verificada formalmente usando la herramienta AVISPA *tool web interface*, la cual provee *agreement* (cuarto nivel de autenticación de acuerdo con la jerarquía de Lowe [19], donde dicho nivel es el más alto). Después de esta etapa, el cliente está listo para firmar o consultar firmas.

3.2 Firmar documentos digitales y autenticarlos

Firma. Esta etapa empieza una vez que los participantes hayan sido autenticados. El cliente usa la llave de sesión K'_{CS} (obtenida en la etapa de autenticación) para cifrar el *e-document* que enviará a *F*, y los datos personales {*dp*}. El servidor firma el *e-document* usando la fórmula signing(*F*, {*dp*}, *al*, K_{CS}) , el cual usa también su conocimiento inicial y relaciona el nombre de usuario con el algoritmo de esteganografía previamente seleccionado por el cliente y la llave secreta compartida.

Autenticar documentos. Para este paso se debe ejecutar el proceso de consulta como proceso seguro, el cual empieza una vez que la etapa II o III haya sido completada. Un cliente autenticado puede consultar la firma de un documento de cualquier cliente autorizado. El servidor mapea el nombre de usuario *un* con el algoritmo *al* y con la llave compartida K_{CS} para usar la función revealing(F',al',K_{CS}) con el objetivo de revelar los datos personales {*dp*} que se incluyen en el documento enviado F'. El servidor regresa {*dp*} como evidencia. Todas las comunicaciones son cifradas usando la llave de sesión.

4 IMPLEMENTACIÓN DE FADD

Para estructurar el desarrollo se utilizan como referencia los niveles presentados en la sección 2 sobre *host* y *network*.

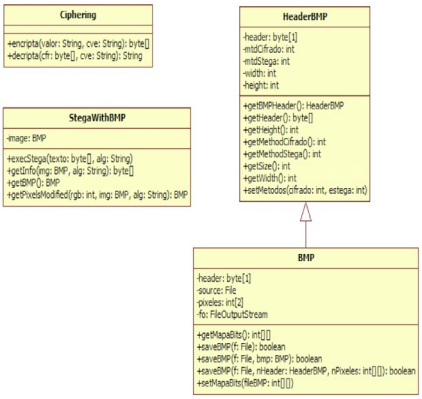

Figura 2. Principales clases UML que forman parte principal de la implementación de FADD de escritorio

4.1 Nivel *host*

En esta parte se implementó el proceso de esteganografía usando la técnica del bit menos significativo LSB con el formato de archivos BMP. La implementación se realizó en Java; esta aplicación puede ser usada para ocultar cualquier tipo de información en modo texto, en modo cifrado o en forma de campos de textos previamente establecidos; usando esta última opción los datos los ocultamos en XML (después de cifrarlo) y esto puede ser usado como protección de derechos de autor. Con respecto a la parte de criptografía se implementaron dos tipos de algoritmos de criptografía simétrica: a) uno basado en AES, y b) el otro basado en *password*. El primero consiste en generar una llave y un *password* por cada usuario; cada *password* se mapea con una llave y se aplica AES con la llave respectiva. El segundo consiste en aplicar una función *hash* a un *password*, la cual se introduce por el usuario, y después se le aplica una función XOR.

Se realizaron, entre otras, cuatro clases principales como puede verse en la figura 2.

- *Ciphering.java*. Cifra y descifra información a partir de cadenas hechas a través de arreglos de bytes y viceversa.
- *HeaderBMP.java*. Lee e interpreta los datos de la cabecera de una imagen BMP.
- *BMP.java*. Guarda, interpreta y manipula datos para la creación de imágenes BMP.
- *StegaWithBMP.java*. Aplica el método esteganográfico de LSB para ocultar información cifrada en una imagen BMP.

4.2 Nivel *network*

De acuerdo con la W3C, un *web service* se define como un software diseñado para soportar interoperabilidad entre máquina y máquina en una red. Esto maneja una interfaz descrita en un lenguaje de descripción de datos que se basa en XML llamada *web service description languaje* (WSDL). Lo que se reporta en esta sección consiste en la implementación de los servicios que provee el sitio web y aunque no están descritos sobre WSDL sí son servicios que provee el sitio web a través de los métodos POST o GET. Dicho sitio se desarrolló en una intranet. Esta implementación usa JavaScript, JSP, PHP y MySQL. La figura 3 muestra la pantalla inicial del sitio web.

A continuación se presentan los servicios provistos por el sitio web.

Conocimiento inicial. En el servidor se almacena una lista de llaves compartidas en una base de datos. Las llaves son generadas en el lado del servidor después que el cliente registra sus datos y completa los pasos en una autenticación basada en email.

Autenticación. Se han realizado algunas estrategias con JavaScript o con Firma de Applets para la parte del cliente y se decidió usar como algoritmo de encriptación base AES, el cual será codificado en JSP para la parte del servidor.

La alternativa actual por la que se ha adoptado fue implementar con JavaScript el algoritmo basado en *password*. Así, el cliente codifica la información compartida con el servidor. El mismo algoritmo se implementa en el servidor.

Figura 3. Firmador y autenticador de documentos digitales versión web

Proceso de firma. Para realizar el proceso de firma y consulta, se utilizan las clases descritas en la figura 2. También se ha implementado una alternativa en donde los clientes pueden verificar firmas de cualquier cliente (sin necesidad que sean o no autorizados), sin embargo, esto es susceptible a ataques, ya que toda la información viaja en texto plano. Este proceso se lleva a cabo siguiendo los siguientes pasos: el usuario sube una imagen al servidor, escribe la información por ocultar, elige las opciones de contraseña, métodos de cifrado y esteganografía. El FADD inserta la firma en la imagen BMP; el usuario puede descargar la imagen firmada.

Proceso de consulta. Éste es el proceso inverso de firma, consiste en leer la información de autenticidad de una imagen firmada y se logra a través de los siguientes pasos: el usuario sube una imagen al servidor, especifica la contraseña y métodos de ocultamiento de los datos, el FADD recupera la información oculta de la imagen con los datos del usuario y arroja el resultado al usuario.

5 TRABAJO RELACIONADO

YusukeLim y colaboradores (2001) [9] propusieron un autenticador de imágenes basado en web, usando marcas de aguas invisibles, conocidas como frágiles. Cualquier usuario autorizado puede generar una imagen con una marca de agua, también puede verificar la autenticidad de la imagen. Intentaron usar *secure socket layer* (SSL) para asegurar acceso al servidor, pero fue dejado como trabajo futuro.

Dicho trabajo tiene como objetivo detectar marcas de agua en un *e-document* aún cuando éste haya sido modificado. A diferencia de eso, el trabajo aquí presentado considera que si un documento se modifica, su autenticidad se pierde ya que su integridad se está violando. En este caso, se debe generar un mensaje de advertencia.

Un número considerable de protocolos de marcas de agua han sido propuestos para proteger al vendedor (dueño de un contenido digital) más que al comprador. Sin embargo, la mayoría ignora la justicia de parte de los clientes. Cuando no se tienen mecanismos apropiados para determinar al distribuidor de una copia pirata, se le conoce como el problema de los derechos del cliente (en inglés *customer's right problem*), primero identificado por Qiao y Nahrstedt [10].

Memon y Wong propusieron un protocolo de marcas de agua denominado *buyer-seller* (comprador-vendedor) [11] para afrontar el problema de los derechos del cliente. En su solución introdujeron un nuevo término: *the unbinding problem*, que consiste en implementar mecanismos de seguridad a un contenido digital para evitar que el vendedor pueda trasplantar una marca de agua y culpar al comprador de una distribución pirata.

Otros autores que también se enfocaron en los problemas *customer's right problem* y *unbinding problem* son Chin-LaungLei y colaboradores [8], quienes enmiendan el esquema propuesto por Memon y Wong [11].

Por su parte, Franco Frattolillo y colaboradores [7] presentaron un protocolo de marcas de agua que puede ser adoptado en un contexto web. Su principal idea es asegurar la protección de derechos de autor de contenido digital distribuido por internet, su trabajo se basó sobre el trabajo de Chin-LaungLei y colaboradores [8].

Todos estos trabajos se basan sobre criptografía asimétrica, el cual requiere infraestructura de llave pública.

En trabajos más recientes, Franco Frattolillo ha mejorado su protocolo de tal forma que un cliente puede ser autenticado con el servidor sin necesidad de que cuente con un certificado digital [6].

Mientras los protocolos mencionados arriba tratan de abordar los dos problemas mencionados (*the customer's right problem* y *the unbinding problem*), nuestro caso no es así, pero tiene muchas similitudes. Adicionalmente, se ha identificado que ninguno de los protocolos mencionado previamente ha sido verificado formalmente. Es por ello que la implementación de este trabajo se ha basado sobre el trabajo presentado por López-Pimentel y colaboradores [5].

6 CONCLUSIONES Y TRABAJO FUTURO

Este artículo describe una implementación de FADD en web que sirve para firmar y autenticar documentos digitales. Se implementa un protocolo de seguridad bajo el esquema de criptografía de llave simétrica. Esta investigación aún está en proceso y como trabajo futuro se está planeando realizar los siguientes puntos:

- Implementar mecanismos de seguridad más robustos, agregar criptografía de llave pública utilizando protocolos estándares como SSL y TLS.
- Para obtener autenticación en el sitio web los participantes deben tener una llave compartida con el servidor. El desafío que se tiene es la distribución de la llave compartida.
- Para que un agente pueda conocer la firma de un documento digital, tiene que saber el algoritmo de esteganografía, el algoritmo criptográfico, las contraseñas, etcétera. Actualmente se está trabajando en mecanismos que permitan incluir esa información redundante a partir de la cabecera de un archivo y que así el cliente no tenga que escribirla (en este caso no necesita hacerlo).
- Establecer mecanismos que permitan a un servidor determinar que un documento digital ya ha sido firmado previamente, cuando un cliente deshonesto quiera refirmar un documento.
- Extender los servicios del sitio web para que sean compatibles con el lenguaje WSDL que permiten publicar *web service*.
- Aún cuando ya se han realizado algunas pruebas de desempeño utilizando una simulación de accesos concurrentes utilizando *threads*, se está considerando realizar pruebas con usuarios reales.

REFERENCIAS

1. Kostopoulos, I., Gilani, S. A. M. Skodras, A. N. (2002). Colour image authentication based on a self-embedding technique. DSP 2002, *14th International Conferenceon Digital Signal Processing*, 2 (1): 733-736.
2. Potdar, V. M., Chang Song Han, E. (2005). Survey of digital image water-marking techniques. *3rd International Conference on Industrial Informatics* (INDIN 2005), 709-716.
3. Documento del Comité de Agricultura de las Naciones Unidas. Disponible en: ftp://ftp.fao.org/docrep/FAO/Meeting/011/j9289e.pdf
4. Indicadores Biodiversidad y Bosques. World watch Institute. Disponible en: url: http://www.ecoestrategia.com/articulos/indicadores/indicadores.html#07
5. López-Pimentel, J. C., Monroy, R. (2008). Formal support to security protocol development: a survey. J*ournal Computación y Sistemas*, special volume 50 Years of Computing in Mexico, 12: 89-108.
6. Frattolillo. F. (2007). Watermarking protocol for web context. *IEEE Transactions on Information Forensics and Security*, 2 (3-1): 350-363.
7. Frattolillo, F., D'Onofrio, S. (2005). A web oriented watermarking protocol. En: *IEC* (Praga), 91-96.
8. Chin-Laung Lei, Pei-Ling Yu, Pan-Lung Tsai, Ming-Hwa Chan (2004). An efficient and anonymous buyer-seller watermarking protocol. ImageProcessing. *IEEE Transactions*, 13: 1618-1626.
9. Yusuk Lim, ChangshengXu, Dagan Feng, D. (2001). Web based image authentication using invisible fragile watermark. En: *Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing*. Volume 11, VIP '01: 31-34, Darlinghurst, Australia. Computer Society, Inc.
10. Lintian Qiao, Nahrstedt, K. (1998). Watermarking schemes and protocols for protecting rightful ownership and customer's rights. J. Vis. Commun. Image Representation, 9: 194-210.
11. Memon, N., Wong, P. W. (2001). A buyer-seller watermarking protocol. *IEEE Trans. Image Processing*, 10: 643-649.
12. Diffie, W., Hellman, M. (1976). New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22 (6): 644-654.
13. Donovan Artz. (2001). Digital steganography: Hiding data within data. *IEEE Internet Computing*, 5: 75-80.
14. Kessler, G. C., Ciphers, N. (2004). An overview of steganography for the computer forensics examiner. Federal Bureau of Investigations, viewed 28.
15. Potdar, V. M., Han, S., Chang, E. (2005). A survey of digital image watermarking techniques. INDIN '05. *3rd IEEE International Conference on Industrial Informatics,* 1 (1): 709-716.
16. Lawrence C. Paulson, L. C. (1998). The inductive approach to verifying cryptographic protocols. *Journal in Computer Security*, 6 (1-2): 85, 128.
17. López-Pimentel, J. C., Monroy, R., Ramos-FonBon, V. F. (2011). Symmetric cryptography protocol for signing and authenticating digital documents. En: *The ICDIPC2011 proceedings, Communications in Computer and Information Science*, Springer LNCS, 9-23.
18. Lowe, G. (1995). An attack on the need-ham-schroeder public-key authentication protocol. InformationProcessingLetters, 56 (3): 131-133.
19. Lowe, G. (1997). A hierarchy of authenti-cation specifications. En: *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, CSFW'97, 31-44. IEEE Computer Society.

Acerca de los autores

Luis Enrique Cuevas Díaz es Ingeniero en Desarrollo de Software por la Universidad Politécnica de Chiapas. Participa como desarrollador en el proyecto de FADD. Tiene estudios en diseño gráfico para el manejo de Corel Draw, Adobe Photoshop, Adobe Dreamweaver, entre otros. Cuenta con estudios de inglés en la escuela de lenguas de la UNACH. Está certificado en un curso de Adobe Flash básico-avanzado. Actualmente participa en un curso de certificación en Java.

Juan Carlos López Pimentel se graduó con mención honorífica de la carrera de Ingeniería en Sistemas Computacionales, en el Tecnológico de Tuxtla Gutiérrez (2001). Posteriormente, obtuvo el grado de Maestro en Ciencias Computacionales en el Tecnológico de Monterrey, Campus Estado de México (ITESM-CEM) (2003); y de Doctor en Ciencias Computacionales (ITESM-CEM, 2008), su investigación fue corrección de protocolos de seguridad con vulnerabilidades. Desde 2011 es miembro del SNI-CONACyT, tiene actualmente el reconocimiento de nivel candidato. Es miembro honorifico del Sistema Estatal de Investigadores del COCYTECH. Ha impartido cursos de nivel licenciatura, maestría y doctorado, destacan los cursos de seguridad informática, sistemas distribuidos, sistemas de información y programación avanzada. Actualmente, es profesor investigador de tiempo completo de la Universidad Politécnica de Chiapas y líder del Cuerpo Académico Sistemas Mecatrónicos Inteligentes, su área de interés es la seguridad computacional enfocado a implementar métodos para hacer más robustos los sistemas.

Rodolfo Esteban Ibarra Orozco es Ingeniero en Comunicaciones y Electrónica por el Instituto Politécnico Nacional. Realizó la Maestría en Ciencias de la Computación y el Doctorado en Ciencias Computacionales en el Tecnológico de Monterrey, Campus Estado de México (2009). Su especialidad es inteligencia artificial y sus áreas de interés son procesamiento de señales y reconocimiento de patrones. Ha participado en la industria realizando análisis de minería de datos a empresas del sector financiero y de prestación de servicios de TI. Actualmente es profesor de tiempo completo de la Universidad Politécnica de Chiapas.

Víctor Fernando Ramos Fon Bon es egresado con mención honorífica de la Universidad Autónoma de Guadalajara, posteriormente realizó estudios de Maestría en Ciencias en el área de Desarrollo de Software en la University of Houston Clear Lake. Tiene más de 20 años de experiencia en desarrollo de software y centros de datos tanto en el país como en el extranjero, entre sus últimas actividades destaca su participación como senior database manager / senior developer con EPIC Interactive Technologies, donde participó en proyectos con Lockheed Martin, NASA, Encompass; con la Universidad Autónoma de Guadalajara se desempeñó como gerente de ingeniería de software y centro de datos; también es asesor externo de una empresa que desarrolla aplicaciones orientadas a proporcionar servicios e insumos a las agencias federales del gobierno de EEUU. Actualmente es profesor-investigador de la Universidad Politécnica de Chiapas, donde desarrolla actividades de Investigación. Sus intereses de investigación son la relación de las ciencias computacionales y la educación así como el procesamiento de imágenes y seguridad.